Volume 6, Issue 9, September 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Implementation of i-vector Algorithm in Speech Emotion Recognition by using Two Different Classifiers: Gaussian Mixture Model and Support Vector Machine

Joan Gomes and Mohamed El-Sharkawy
Department of Electrical & Computer Engineering, Indiana University-Purdue University Indianapolis (IUPUI)
Indianapolis, IN 46202, USA

Abstract: Emotions constitute an essential part of our existence as it exerts great influence on the physical as well as mental health of people. Speech is considered as the most powerful mode to communicate with intentions and emotions. Over the past few decades a great deal of research has been done to recognize human emotion using speech information. Many systems have been proposed to make the Speech Emotion Recognition (SER) process more correct and accurate. This paper discusses the design of a speech emotion recognition system implementing a comparatively new method- i-vector model. i-vector model has found much success in the areas of speaker identification, speech recognition and language identification. But it has not been much explored in recognition of emotion. In this research i-vector model was implemented in processing extracted features for speech representation. Two different classification schemes were designed using two different classifiers - Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) along with i-vector algorithm. Performance of these two systems was evaluated using the same emotional speech database to classify four emotional states: anger, happiness, sadness and neutral. Results were compared and more than 75% of accuracy was obtained by both of these systems which proved that our proposed i-vector algorithm can identify speech emotions with less error and more accuracy.

Keywords: Speech Emotion Recognition (SER), Gaussian Mixture Model (GMM), GMM Universal Background Model (UBM), Maximum A Posteriori (MAP) Adaptation, i-vector Algorithm, Support Vector Machine (SVM), Formant Frequency, Mel Frequency Cepstrum Coefficients (MFCC)

I. INTRODUCTION

Emotions exert an incredibly powerful force on human behavior. In psychology, emotion is often defined as a complex state of feeling that results in physical and psychological changes that influence thought and behavior [1]. Speech emotion analysis refers to the use of various methods to analyze vocal behavior as a marker of state of the speaker (e.g. emotions, moods, and stress). The basic assumption is that there is a set of objectively measurable voice parameters that reflects the affective state a person is currently experiencing and these parameters get modified depending on different emotional states during the voice production process [2]. With the advancements of technologies, a good number of researches have been done on emotion in the field of psychology and physiology. Anger, fear, disgust, sadness, surprise, happiness - were six basic types of emotions detected in early stage. Amusement, contempt, contentment, embarrassment, excitement, guilt, pride in achievement, relief, satisfaction, sensory pleasure, shame – these emotions were included later. Speech Emotion Recognition (SER) is quite new but a quickly growing field in the vast area of digital signal processing because of its notably immense application in different areas of modern life. Analysis of emotion in speech can be extremely useful in developing communication systems for vocally-impaired individuals or for autistic children. Its most important application is in intelligent human-machine interaction. Other applications of Speech Emotion Recognition (SER) include robotics, psychiatric diagnosis, health care, lie detection, intelligent toys, learning environment, children education, educational software, dialog systems, call centers, security fields, and entertainment [3].

II. EMOTION RECOGNITION FROM SPEECH

Speech emotion recognition aims to automatically identify the emotional state of a human being from his or her voice. It is based on in-depth analysis of the generation mechanism of speech signal, extracting some features which contain emotional information from the speaker's voice, and taking appropriate pattern recognition methods to identify emotional states [4]. So the complete process of synthesizing speech and then decoding and identifying emotions is a complex task. Usually this can be executed in three steps:

1) Speech Signal Database - The primary requirement when investigating speech emotions is to choose a valid database, which is going to be the basis of the subsequent research work. Throughout the world English, German, Spanish, and Chinese single language emotion speech databases have been built. A few speech libraries also contain a variety of languages. Some examples of emotion speech database are: EMO-DB, AIBO, CSLO, and BUAA [5].

2) Feature Extraction - Choosing suitable speech features for developing an emotion recognition system is also crucial. Mainly three types of features are extracted from speech [6].

Table 1. Types of Features Representing Speech

Frequency Characteristics	**Time-related Features**	**Voice Quality Parameters and Energy Descriptors**
Accent shape, Average pitch, Contour slope, Final lowering, Pitch range	Speech rate, Stress frequency	Breathiness, Loudness, Pause discontinuity, Pitch discontinuity, Brilliance

3) Identifying Emotion (Training, Testing & Classifying) - Different statistics based mathematical models and stochastic processes are applied to train, test and classify the speech samples. Accuracy rate of speech emotion recognition are different for different models [6]. Some commonly used statistical models are:

- Linear Discriminant Classifiers (LDC)
- K Nearest Neighbors (k-NN)
- Gaussian Mixture Model (GMM)
- Support Vector Machine (SVM)
- Artificial Neural Networks (ANN)
- Decision Tree Algorithms
- Hidden Markov Models (HMM)
- Deep Belief Network (DBM)

III. THEORETICAL CONCEPTS

3.1. Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) is a weighted sum of M component Gaussian densities as given by the equation,

$$p(x|\lambda) = \sum_{i=1}^{M} w_i \, g(x|\mu_i, \sum_i) \quad (1)$$

Where x is a D-dimensional continuous-valued data vector (i.e. measurement of features), $w_i, i = 1, \dots, M,$ are the mixture weights, and $g(x|\mu_i, \sum_i), i = 1, \dots, M,$ are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form,

$$g(x|\mu_i, \sum_i) = \frac{1}{2\pi^{D/2} |\sum_i|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_i)' \sum_i{}^{-1} (x-\mu_i)\} \quad (2)$$

with mean vector μ_i and covariance matrix $\sum_i$. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\lambda = \{w_i, \mu_i, \sum_i\} \quad i = 1, \dots, M \quad (3)$$

GMMs are capable of representing a large class of simple distributions. One of the powerful attributes of the GMM is its ability to form smooth approximations to arbitrarily shaped densities. GMM not only provides a smooth overall distribution fit, its components also clearly detail the multi-modal nature of the density. GMMs are widely used in speech emotion recognition systems, as it can easily be used as a parametric model of the probability distribution of continuous measurements of features such as vocal-tract related spectral features in a speech processing system [7, 8].

3.2. Universal Background Model (UBM)

The Universal Background Model (UBM) is a large GMM trained to represent the distribution of features extracted from different speech samples. In the GMM-UBM system a single, independent background model is used to represent $p(x|\lambda)$ derived from (1). This hypothesized background model is derived by adapting the parameters of the UBM using the speech sample data and a form of Bayesian Adaptation. Speech samples which reflect the expected alternative speech to be encountered during emotion recognition are selected. There is no objective measure to determine the right number of speakers or amount of speech to use in training a UBM. Given the data to train a UBM, there are many approaches that can be used to obtain the final model. The simplest is to pool all the data to train the complete UBM. The pooled data should be balanced over the subpopulations within the data. For example, in using speech samples for emotion recognition one should be sure that there is a balance of all different emotion categories. Otherwise, the final model will be biased toward the dominant emotion category [8]. Gaussian mixture models with universal backgrounds (UBMs) have become the standard method for speech signal analysis. Typically, a speaker model is constructed by Maximum *A Posteriori* (MAP) adaptation of the means of the UBM. A GMM super vector is constructed by stacking the means of the adapted mixture components [9].

3.3. Maximum A Posteriori (MAP) Parameter Estimation

Maximum *A Posteriori* (MAP) estimation is used to estimate the GMM parameters. The MAP estimation is a two-step estimation process. In first step estimates of the sufficient statistics of the training data are computed for each mixture in the prior model. In second step these "new" sufficient statistic estimates are then combined with the "old" sufficient statistics from the prior mixture parameters using a data-dependent mixing coefficient. The data-dependent mixing coefficient is designed so that mixtures with high counts of new data rely more on the new sufficient statistics for final parameter estimation and mixtures with low counts of new data rely more on the old sufficient statistics for final parameter estimation.

Given a prior model and training vectors from the desired class, X = $\{x_1, x_2, \ldots, x_T\}$, first the probabilistic alignment of the training vectors into the prior mixture components are determined. That is, the sufficient statistics for the weight, mean and variance parameters are computed.

$$n_i = \sum_{t=1}^{T} Pr\left(i \middle| x_t, \lambda_{prior}\right) \quad \text{(Weight)} \tag{4}$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} Pr\left(i \middle| x_t, \lambda_{prior}\right) x_t \quad \text{(Mean)} \tag{5}$$

$$E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} Pr\left(i \middle| x_t, \lambda_{prior}\right) {x_t}^2 \quad \text{(Variance)} \tag{6}$$

The adaptation coefficients controlling the balance between old and new estimates are $\{{\alpha_i}^w, {\alpha_i}^m, {\alpha_i}^v\}$ for the weights, means and variances, respectively. This is defined as

$${\alpha_i}^\rho = \frac{n_i}{n_i + r^\rho} \quad \rho \in \{w.m.v\} \tag{7}$$

where r^ρ is a fixed "relevance" factor for parameter ρ. Lastly these new sufficient statistics from the training data are used to update the prior sufficient statistics for mixture *i* to create the adapted parameters for mixture *i* with the equations:

$$\widehat{w_i} = \left[\frac{{\alpha_i}^w n_i}{T} + (1 - {\alpha_i}^w) w_i\right] \gamma \tag{8}$$

$$\widehat{\mu_i} = {\alpha_i}^m E_i(x) + (1 - {\alpha_i}^m) \mu_i \tag{9}$$

$$\widehat{{\sigma^2}_i} = {\alpha_i}^v E_i(x^2) + (1 - {\alpha_i}^v)({\sigma^2}_i + {\mu_i}^2) - \widehat{\mu_i}^2 \tag{10}$$

where the scale factor, γ, is computed over all adapted mixture weights to ensure they sum to unity.

MAP estimation is used in speaker recognition applications to derive speaker model by adapting from a universal background model (UBM). For example, Fig. 1 and Fig. 2 show two steps in adapting a hypothesized speaker model. In Fig. 1 the training vectors are probabilistically mapped into the UBM (prior) mixtures. In Fig. 2 the adapted mixture parameters are derived using the statistics of new data and the UBM (prior) mixture parameters.

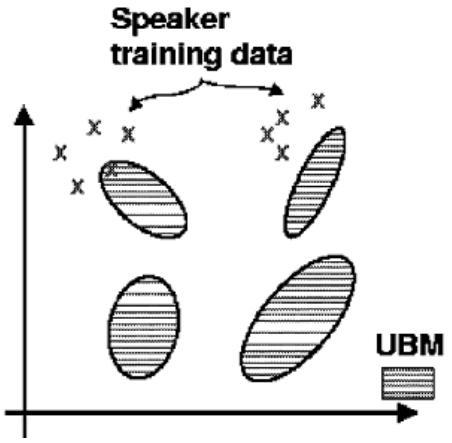

Figure 1. MAP Adaptation step 1

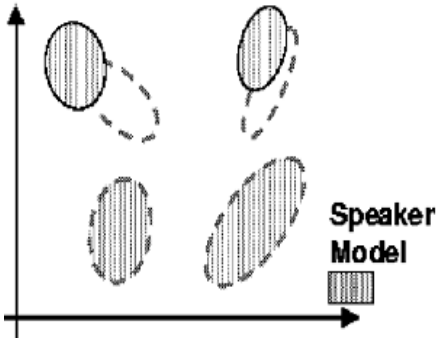

Figure 2. MAP Adaptation step 2

MAP is also used in other pattern recognition tasks where limited labeled training data is used to adapt a prior, general model [7, 8].

3.4. Support Vector Machine (SVM)

Support Vector Machine (SVM) is an effective approach for pattern recognition. Here, the basics of the SVM will be presented briefly. In SVM approach, the main aim of an SVM classifier is obtaining a function f(x), which determines the decision boundary or hyperplane. This hyperplane optimally separates two classes of input data points [10]. This hyperplane is shown in Fig.3. So the main idea of SVM [11, 12] is to transform the original input set to a high dimensional feature space by using a kernel function, in which input space consisting of input samples is converted into high dimensional feature space and therefore the input samples become linearly separable. It is clearly explained by using an optimal separation hyperplane in Fig. 3.

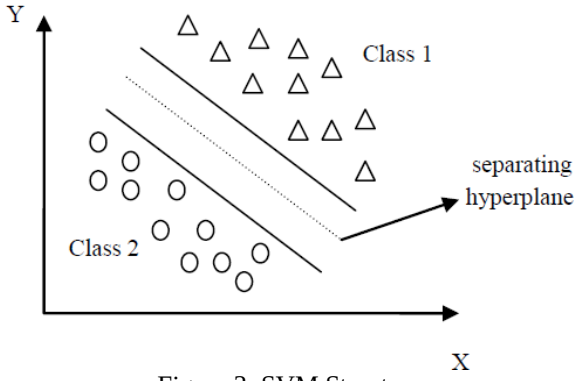

Figure 3. SVM Structure

The main advantage of SVM is that it has limited training data and hence has very good classification performance. For linearly separable data points, classification is done by using the following formula [13],

$\langle w,x \rangle + b_0 \geq 1$ (11)

$\langle w,x \rangle + b_0 \geq -1$ (12)

Where (x,y) is the pair of training set. Here, x∈R and y ∈{-1,+1}. ‹w,x› represents the inner product of w and x whereas b_0 refers to the bias condition. SVM that employs both the linear kernel function and the Radial Basis Kernel (RBF) function is used here. The linear kernel function is given by the formula below,

Kernel (x,y) = (x . y) (13)

The radial basis kernel function is given by the following formula,

Kernel(x,y) = $e^{\frac{-||x-y||^2}{2\sigma^2}}$ (14)

The SVM classifier places the decision boundary by using maximal margin among all possible hyper planes [14].

The Support Vector Machine (SVM) is widely used as a classifier for emotion recognition. The SVM is used for classification and regression purpose. It performs classification by constructing an N-dimensional hyperplane that optimally separates data into categories. The classification is achieved by a linear or nonlinear separating surface in the input feature space of the dataset [15].

3.5. i-vector Algorithm

The conventional i-vector extraction is a probabilistic compression process which reduces the dimensionality of the vectors. It models the super vector $M_{(s,h)}$ as the sum of the independent mean vector m and total variability vector.

$M_{(s,h)} = m + T\, w_{(s,h)}$ (15)

where m is the super vector, T and $w_{(s,h)}$ represents the total variability matrix and i-vector respectively. Extraction of i-vector will minimize the variability and will normalize the co-variance of super vectors [16].

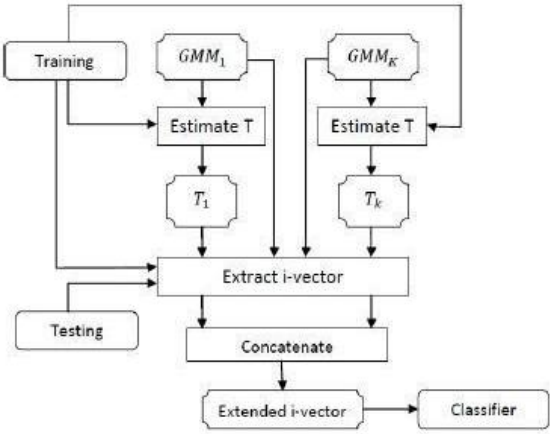

Figure 4. i-vector Algorithm Model

Fig. 4 shows i-vector algorithm model. First super vector is trained using neutral based corpus (GMM_g in Fig. 4) and emotion specific SVMs are trained by MAP adaption (GMM_1 in Fig. 4). After that i-vector features are generated for different emotional specific GMMs which are then concatenated to form extended i-vector features [17].

IV. EXPERIMENT

4.1. Speech Database

For our study the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database collected at Signal Analysis and Interpretation Laboratory (SAIL) at University of Southern California (USC) was used [18]. IEMOCAP database is an acted, multimodal and multi speaker database. A total of 11.5GB of data contains 12 hours of both improvised and scripted sessions of 10 actors (male & female). The database contains 4 types of emotion speech samples- angry (25%), happy (15%), sad (20%) and neutral (40%).

4.2. Feature Extraction

The input to the system is a .wav file from IEMOCAP database that contains emotional speech utterance from different emotional classes. After that feature extraction process is carried out. A total of 51 features were extracted from each speech sample using OpenSMILE toolkit. OpenSMILE toolkit is a modular and flexible feature extractor for signal processing specifically for audio-signal features. It is written purely in C++ and capable of data input, signal processing, general data processing, low-level audio features, functional, classifiers and other components, data output, and other capabilities [19].

Table 2. List of Extracted Features

Features	
Pitch Contour – Minimum, Maximum, Mean	1-3
Formant Frequency – Minimum, Maximum, Mean	4-6
Log Energy (LE) - Minimum, Maximum, Mean	7-9
Average Magnitude Difference (AMD) -Minimum, Maximum, Mean	10-12
Mel-Frequency Cepstral Coefficients (MFCC)	13-25
MFCC (1st Derivative)	26-38
MFCC (2nd Derivative)	39-51

Formant Frequencies are the resonant frequencies of the vocal tract. Speech scientists described formants as quantitative characteristics of the vocal tract since the location of vocal tract resonances in the frequency domain, depends upon the shape and the physical dimensions of the vocal tract [20]. Mel-Frequency Cepstral Coefficients (MFCC) are the coefficients which represent the vocal tract and are widely used in audio analysis & recognition. The 1st & 2nd derivatives of MFCCs demonstrate change over time. MFCCs & derivatives were resorted to easily compare patterns. MFCC in the low frequency region has a good frequency resolution, and the robustness to noise is also very good, but the accuracy of high frequency coefficient is not satisfactory. In our research we extracted the first 12-order of the MFCC coefficients [4]. The process of calculating MFCC is shown in Fig. 5 [15].

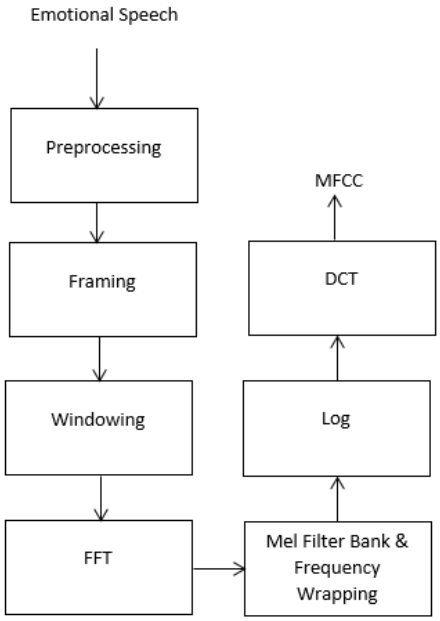

Figure 5. Process of Calculating MFCC

4.3. GMM UBM Calculation and i-vector Extraction

Software used in this step was Matlab, which is a widely used piece of software in the field of identification of human speech components. Matlab contains vast collection of audio signal processing methods. It has an easy-to-use programming and many build-in algorithms for processing speech signals [21]. Extracted features by using OpenSMILE toolkit were used to train and classify every emotion. The GMM model algorithm condenses the 12 features and the 39 MFCCs. Then GMM UBM mixture components were computed for each speech sample using MAP adaptation algorithm. The multi-dimension i-vector of each sample is extracted. The total variability matrix T is trained by all the training speech samples. For conventional i-vector, Linear Discriminant Analysis (LDA) strategy is applied to reduce the dimensionality of i-vectors [22]. Emotion groups were formed based on the average value of the first 12 features and the variance of each MFCCs according to the range of data. Fig.6 shows four emotion groups according to the average frequency values and the variance of MFCC's for different samples [6].

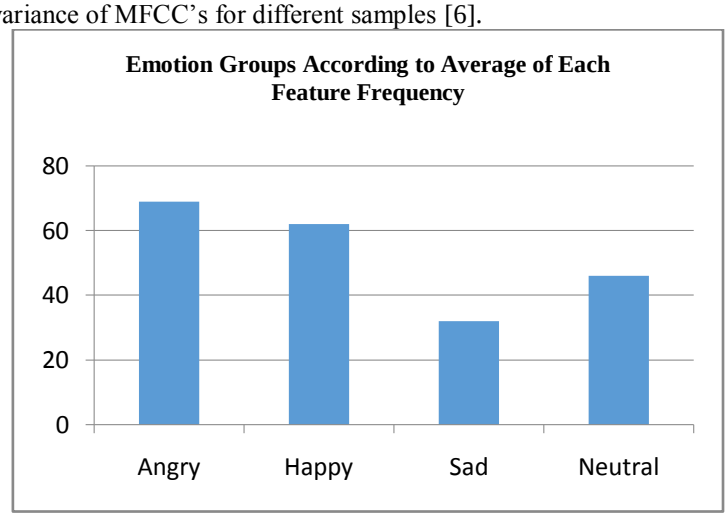

Figure 6. Classification of Emotion Groups

4.4. SVM Classification and i-vector Extraction

All of the calculated features were put into an Nx51 matrix where N is equal to the total number of samples in the input signals. This matrix was given as input to the SVM classifier. The complete system design is shown in Fig. 7 [3].

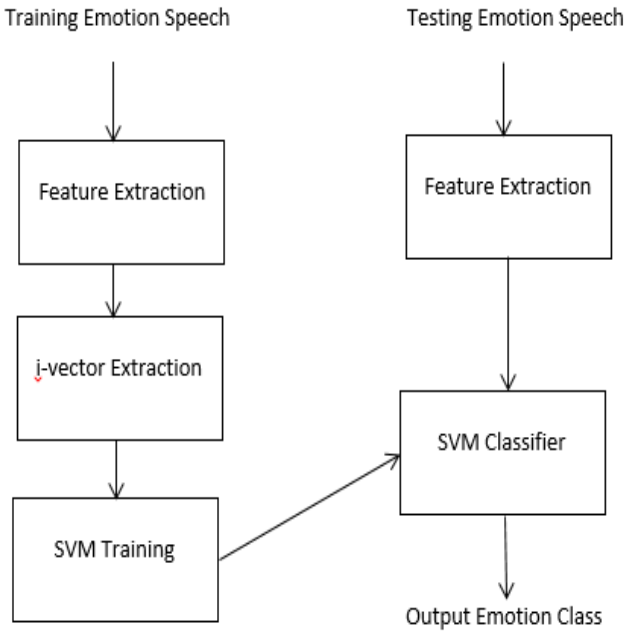

Figure 7. System Diagram

The libsvm tool in Matlab was used to do the cross validation of models and analyzing of results. For each emotion, speech utterances were divided into two subsets as training subset and testing subset. The number of speech utterances for emotion as the training subset is 90% and 10% as the test subset [4]. First the classifier was trained with speech samples from training subset. After training the classifier, it was used to recognize the new given input speech sample from testing subset. The dimensionality and the variability of output of the classifier was reduced by using i-vector extraction method. Final output of the system is a label of a particular emotion

class. There are total four classes- angry, happy, sad and neutral. Each label represents corresponding emotion class. Fig.8 shows four emotion classes:

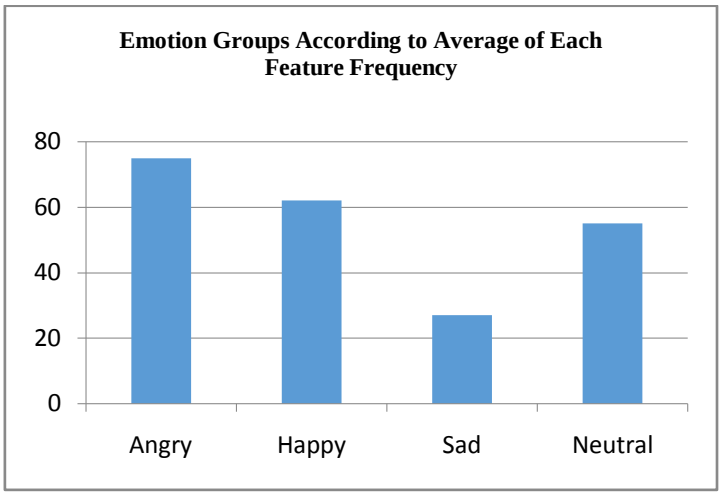

Figure 8. Classification of Emotion Groups

V. RESULT

New input signals were classified based on those emotion groups. Each new input signal's features were compared with each emotion group feature frequency values and were categorized accordingly. Speech signal samples used to train the classifier and to test the classifier were kept different.

The identification rates of the system using only GMM-UBM algorithm and using i-vector algorithm with GMM-UBM algorithm are shown in Table 3.

Table 3. Identification Rate of Emotions

Category	**Only GMM-UBM Algorithm (%)**	**With i-vector Algorithm (%)**
Angry	49.63	63.87
Happy	81.35	90.36
Sad	63.77	78.26
Neutral	54.91	69.68
Average	**62.42**	**75.54**

It can be seen from Table 3 that proposed i-vector algorithm can enhance the performance of emotion recognition in each four emotional state. The average identification rates increase by 21.02% compared with that of conventional GMM-UBM algorithm. Also overall this novel emotion identification system was almost 76% accurate [6].

The identification rates of the system using only SVM algorithm and using i-vector algorithm with SVM algorithm are shown in Table 4 [3].

Table 4. Identification Rate of Emotions

Category	**Only SVM Algorithm (%)**	**With i-vector Algorithm (%)**
Angry	48.15	62.25
Happy	81.35	91.33
Sad	61.97	79.13
Neutral	54.91	74.66
Average	**61.60**	**77.59**

It can be seen from Table 4 that proposed algorithm can identify different types of emotions with good number of accuracy. For happy emotion the identification rate is as high as 92%. When i-vector algorithm was introduced it enhanced the performance of emotion recognition significantly. The average identification rates increase by 25.96% compared with that of conventional SVM algorithm. Also overall this proposed emotion identification system was almost 78% accurate, well above other researchers' results for the same tests.

Fig. 9 shows the graphical representation of our result:

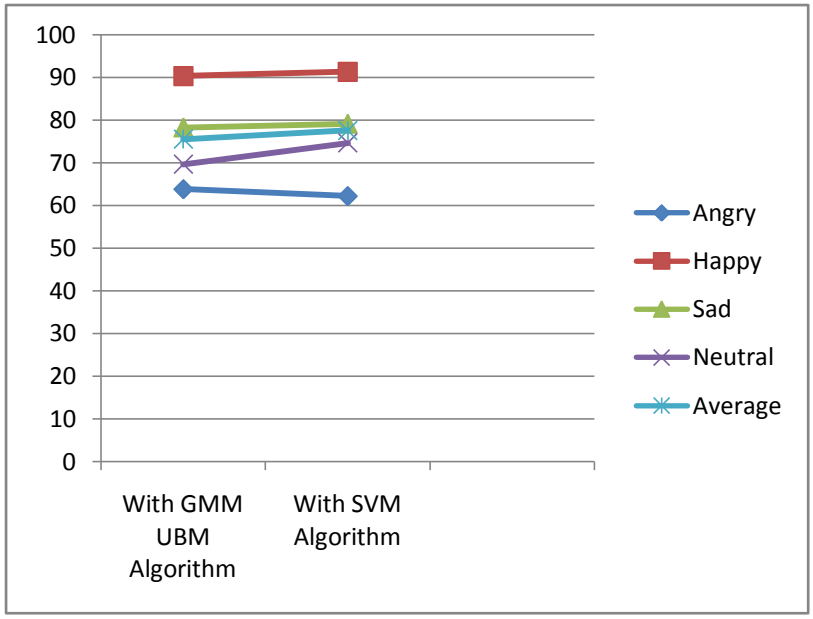

Figure 9. Graphical Representation of Experimental Result

VI. CONCLUSION

From experimentation and result it is proved that we successfully developed, trained, and tested a classification system to identify emotions from speech signals of different emotions. Soon that day will come when a real-time system capable of determining any emotions at a human-comparable accuracy will be established. Emotion recognition has already been introduced for security, gaming, user-computer interactions, and lie detectors. As well, real-time emotion recognition can be of great help to the autistic children to recognize emotions. But currently used emotion recognition systems are often highly inaccurate in realistic settings. Our proposed i-vector algorithm has achieved accuracy of 78% while implemented with two different statistical algorithms which is really good if compared to the other available systems. More work is needed to improve the system so that it can be better used in real-time speech emotion recognition. By our research we successfully established a method for emotion recognition from speech signals which improved the accuracy of speech emotion recognition process statically and dynamically.

REFERENCES

[1] psychology.about.com/od/psychologytopics/a/theories-ofemotion.html
[2] P. N. Juslin, K. R. Scherer, “Speech emotion analysis”, Scholarpedia, 3(10):4240, 2008
[3] J. Gomes, M. El-Sharkawy, “Speech emotion recognition system by using support vector machine and i-vector algorithm”, Interspeech 2016, San Francisco, September 2016
[4] Y. Pan, P. Shen, L. Shen, “Speech emotion recognition using Support Vector Machine”, International Journal of Smart Home, vol. 6, no. 2, April 2012
[5] A. Krishnan, M. Fernandez, “The recognition of emotion in human speech, static and dynamic analysis”, Siemens Competition 2010, September 2010
[6] J. Gomes, M. El-Sharkawy, “i-vector algorithm with Gaussian Mixture Model for efficient speech emotion recognition”, The 2015 International Conference on Computational Science and Computational Intelligence (CSCI), December 2015
[7] D. Reynolds, “Gaussian mixture models”, MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA
[8] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, “Speaker verification using adapted Gaussian mixture models”, Digital Signal Processing 10, 19-41(2000)
[9] W. M. Campbell, D. E. Sturim, D. A. Reynolds, A. Solomonoff, “SVM based speaker verification using a GMM super vector kernel and NAP variability compensation”, MIT Lincoln Laboratory, Lexington, MA 02420
[10] B. Panda, D. Padhi, K. Dash, S. Mohanty, “Use of SVM classifier & MFCC in speech emotion recognition system”, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, issue. 3, March 2012
[11] S.V.N. Vishwanathan, M. N. Murty, “SSVM: A simple SVM algorithm”, Indian Institute of Science, Bangalore, India
[12] J. Watson, “Support Vector Machine tutorial”, NEC Labs America, Princeton, USA
[13] V. K. Ingle, J. G. Proakis, “Digital Signal Processing Using Matlab V.4 (Bk & Disked.)”, Boston, MA: PWS Publishing Company, 1996
[14] A. Milton, S. S. Roy, S. T. Selvi, “SVM scheme for speech emotion recognition using MFCC feature”, International Journal of Computer Applications (0975 – 8887), vol. 69, no. 9, May 2013
[15] Y. Chavhan, M.L. Dhore, P. Yesaware, “Speech Emotion Recognition using Support Vector Machine”, 2010 International Journal of Computer Applications (0975-8887), vol. 1, no. 20

[16] L. Chen, Y. Yang, "Emotional speaker recognition based on i-vector through atom aligned sparse representation", Zhejiang University, College of Computer Science & Technology, Hangzhou, China

[17] Xia, Rui, Yang Liu. "Using i-vector space model for emotion recognition." Thirteenth Annual Conference of the International Speech Communication Association, 2012.

[18] *C. Busso, M. Bulut, C.C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S.S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," Journal of Language Resources and Evaluation, vol. 42, no. 4, pp. 335-359, December 2008.*

[19] audeering.com/research/opensmile.html

[20] A. Jacob, P. Mythili, "Upgrading the performance of speech emotion recognition at the segmental level", IQSR Journal of Computer Engineering (IQSR-JCE), e-ISSN:2278-0661, p-ISSN: 2278-8727, Volume 15, Issue 3 (Nov. – Dec. 2013), PP 48-52

[21] V. K. Ingle, J. G. Proakis, "Digital Signal Processing Using Matlab V.4 (Bk & Disked.)", Boston, MA: PWS Publishing Company, 1996

[22] H. Yu, J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition", Pattern Recognition, 34(2001) 2067-2070

